METHOD AND APPARATUS FOR BEZIER CURVE APPROXIMATION DATA COMPRESSION

FIELD OF THE INVENTION

The present invention generally relates to a method and an apparatus for data compression, more specifically to a method and apparatus for compressing Bezier curve approximation data.

BACKGROUND OF THE INVENTION

Recently, small handheld computing devices, such as personal digital assistants ("PDAs"), have become increasingly popular. Due to their sizes, these handheld computing devices are typically not equipped with full-size keyboards. Some of these handheld computing devices support the use of the full-size keyboards as external attachments, and others offer reduced size keypads. Instead of providing a keyboard as an input interface, however, a typical handheld computing device provides a large display, occupying a substantial proportion of the handheld computing device, which is capable of displaying information as well as being capable of functioning as an input interface. Entering data through the display is generally accomplished by utilizing a writing instrument such as a pen or stylus, and a user typically enters information or data by directly writing on the display using the pen. Resulting hand-drawn objects, such as free-hand drawings, geometric shapes, and handwritten letters and characters, are captured as digital ink, and paths the pen has taken appear on the display. The digital ink represents coordinates and time information of the paths which the pen has taken to produce the hand-drawn objects on the display. Digital ink is typically expressed as poly line objects in series of pen points (x,y,t).

A small computing device utilizing digital ink typically samples points along each path the pen has taken at a predetermined frequency to approximate the path and

the object drawn. However, data of the sampled points contain large amount of redundancy and can quickly become very large, which requires a large memory in the small computing device to store the data. If the user then wishes to transfer the data to another device, the transfer time may be unreasonably long due to the size of the date and the transfer rate available to the small computing device. This transfer time duration becomes even more apparent considering that due to the mobile and portable nature of the small computing device, the small computing device is likely to be connected to another device through a wireless network such as a cellular network, which has a relative narrow bandwidth, or a low rate of data transfer. To reduce the size of the data, the small computing device may perform data compression on the data of the sampled points, and then transmit the compressed data to the other device through the available network. The compressed data received by the other device is decompressed to recreate the object originally drawn. If the decompressed data is the same as the original data of the sampled points, then the compression-decompression process is said to be lossless; that is, no information has been lost due to the compression-decompression process. However, if the decompressed data is different from the original data of the sampled points, even slightly, then the compression-decompression process is said to be lossy; that is, some information has been lost due to the compression-decompression process.

Although a lossless process is preferred for an accurate representation, it is often unnecessary to reproduce the drawn object with the accuracy of the lossless process. It is often more desirable to reduce the data size for a reasonable and adequate representation of the object drawn than to maintain the large data size for the accurate reproduction of the drawn object.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is an exemplary electronic device displaying a hand-drawn object;

FIG. 2 is an exemplary electronic device displaying a Bezier curve approximation of the hand-drawn object and Bezier control points;

FIG. 3 is an exemplary device displaying a representation of the hand-drawn object using an exemplary data compression process and Bezier control points;

FIG. 4 is a flowchart illustrating an exemplary process of compressing digital ink;

FIG. 5 is an exemplary illustration of a visual comparison between the hand-drawn object and the final representation of the hand-drawn object;

FIG. 6 is a flowchart illustrating an exemplary process of dividing digital ink into smaller digital ink strokes;

FIG. 7 is an exemplary illustration of Bezier curve and straight line approximations;

FIG. 8 is a flowchart illustrating an exemplary process of determining whether a Bezier curve can be adequately represented by a straight line;

FIG. 9 is a flowchart illustrating an exemplary process of modifying Bezier control points for further reduction in data size; and

FIG. 10 is an exemplary block diagram of an electronic device according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention generally relates to a method and an apparatus for data compression, more specifically to a method and apparatus for compressing Bezier curve approximation data. An object, typically a hand-drawn object on a display of an electronic device such as a PDA, is a collection of points on the display, and the collection is referred as digital ink. The digital ink representing the object is first broken into smaller segments, or strokes, and each of these strokes is then approximated by a Bezier curve, which comprises Bezier control points. Each Bezier curve is then examined to determine whether it can be adequately represented by a straight line based upon a first predetermined condition. The Bezier curves, which meet the first predetermined condition, are replaced by straight lines. The Bezier curves, which fail to meet the first predetermined condition, are then examined to determine whether they can be better represented by modifying the corresponding Bezier control points. The Bezier curves, which fail to meet the first predetermined condition but are determined to be better represented by modifying the corresponding Bezier control points, receive modified Bezier control points.

FIG. 1 is an exemplary electronic device 100 having a display 102 displaying a hand-drawn object 104, which is a representation by digital ink comprising digital ink points. FIG. 2 is the electronic device 100 displaying a Bezier curve approximation 202 of the hand-drawn object 104 on the display 102, and Bezier control points (only six Bezier control points are indicated, 204, 206, 208, 210, 212, and 214). FIG. 3 is the electronic device 100 displaying a final representation 302 of the hand-drawn object 104 on the display 102 using the presently described method, and Bezier control points (only six Bezier control points are indicated, 304, 306, 308, 310, 312, and 314).

FIG. 4 is a flowchart 400 illustrating an exemplary process of compressing digital ink. The process begins in block 402, and a delta size for the digital ink points is selected in block 404. The delta size is used to select consecutive points for approximating the digital ink. In block 406, the digital ink representing the entire hand-drawn object 104 is divided into a set of smaller digital ink strokes, and then each of the digital ink strokes is approximated by using a quadratic Bezier curve

approximation in block 408. As a result of this approximation step, each digital ink stroke is now approximated and represented by a Bezier curve having Bezier control points. The Bezier control points are two on-line points where a corresponding approximated line goes through and at least one off-line point where curvature of the corresponding approximated line is controlled. As previously described, the resulting approximation 202 is shown in FIG. 2. At this point, a data size of a first compressed data representing the digital ink based upon the quadratic Bezier curve approximation may be calculated.

Each Bezier curve is then examined to determine whether it can be adequately represented by a straight line in block 410. The Bezier curves that are determined to be adequately representable by straight lines are grouped into a first group of the Bezier curves in block 412, and the Bezier curves of the first groups are re-represented with straight lines to approximate the corresponding digital strokes in block 414. The Bezier curves that are determined not to be adequately representable by straight lines are grouped into a second group of the Bezier curves in block 416. The Bezier control points of the Bezier curves of the second group are evaluated and modified in block 418. Resulting first and second groups are then combined to determine a data size of a second compressed data in block 420. The second compressed data size is then compared to the first compressed data size to determine whether a desired compression is achieved in block 422. If the desired compression is achieved, then the process terminates in block 424. As previously described, the resulting final approximation 302 is shown in FIG. 3. However, if the desired compression is not yet achieved, then a new delta size is selected in block 426, and the process is repeated from block 406. After the desired compression is achieved, the second compressed data may be losslessly compressed for further activities such as storage in memory and transmission to another device. FIG. 5 is an exemplary illustration of a visual comparison between the originally drawn object 104 and the final approximation 302, which is overlaid on the originally drawn object 104.

FIG. 6 is a flowchart illustrating an exemplary process of dividing digital ink into smaller digital ink strokes of block 406. In block 602, curvature at each digital ink point within a predetermined size window is estimated by averaging all curvature values at each digital ink point within the predetermined size window. In block 604,

the estimated curvature value is compared against a predetermined threshold curvature value. Because high curvature points such as sharp turns are typically difficult to handle for curve fitting, a digital stroke having an estimated curvature value greater than the threshold curvature value is assumed to contain a sharp turn, and is split into sub-strokes in block 606. The process is repeated from block 602 based on the sub-strokes. If the estimated curvature is determined to be less than the predetermined threshold curvature value, then the digital ink stroke is assumed to be smooth enough, and the process continues to block 408.

After each stroke is represented by the Bezier curve with Bezier control points in block 408, fitting error and dynamic range of the control points may be checked. The Euclidian distance between actual ink point and corresponding point of the Bezier curve is assumed to be the fitting error. The dynamic range of control points is measured by a minimum bit size which can hold the maximum value of difference between consecutive two control points. If the fitting error and the dynamic range are greater than a predetermined acceptable ranges, then new splitting points are determined using a lower curvature threshold, and the process may be repeated from block 602 with the lower curvature threshold.

If the curvature is not large, then a Bezier curve may be adequately represented by a replacement straight line. FIG. 7 is an exemplary illustration of a Bezier curve 700, which is being considered in block 410 for determining whether it can be adequately represented by a straight line 702. The Bezier curve 700 has on-line control points 704 and 706, an off-line control point 708, and an error tolerance boundary 710. The error tolerance boundary is defined by an error tolerance 712, which is a radial distance originating from an on-line control point. A Euclidian distance 714 is shown as the distance between the off-line control point 708 and the straight line 702. FIG. 8 is a flowchart 800 illustrating an exemplary process of block 410 of determining whether the Bezier curve 700 can be adequately represented by a straight line 702. In block 802, the off-line Bezier control point 708 of the Bezier curve 700 is located. In block 804, the error tolerance boundary 710 defined by the error tolerance 712 is identified, and whether the off-line control point 708 is within the error tolerance boundary 710 is determined. If the off-line control point 708 is determined to be within the error tolerance boundary 710, then the process continues

to block 412. If the off-line control point 708 is determined not to be within the error tolerance boundary 710, then the process continues to block 416. For example, the off-line control point 708 may be assumed to be within the error tolerance boundary 710 if the Euclidian distance 714 is less than the error tolerance 712. The Bezier curves having the off-line control points within respective error tolerance boundaries are replaced with straight line representations in block 414.

The Bezier curves determined not have the off-line control points within their respective error tolerance boundaries in block 410, i.e. the Bezier curves that cannot be adequately represented by straight lines, are further examined to reduce the total data size by modifying the on-line and off-line control points in block 418. FIG. 9 is a flowchart illustrating an exemplary process of block 418 of modifying Bezier control points for further reduction in data size. In block 902, each Bezier control point is represented by the corresponding element identification, an X-axis coordinate, a Y-axis coordinate, and a curve status. The element identification identifies which Bezier curve the Bezier control point belongs to, the X-axis and Y-axis coordinates represent the coordinates of the Bezier control point on the display, and the curve status indicates whether the Bezier control point is an on-line control point or an off-line control point. In block 904, the Bezier control points are separated and grouped into X-coordinate array and Y-coordinate array, with each element of the arrays identified by the corresponding element identification and the corresponding coordinate. In block 906, first order difference vectors are calculated for consecutive array elements in each array. Based on the first order differences calculated in block 906, the Bezier control points are separated into a preferred group, which is determined to perform well under entropy compression, and a non-preferred group in block 908. This determination may be made by re-representing the first order difference vectors of each array into magnitude and sign vectors, calculating a histogram of the magnitude vectors in each array, and keeping a certain predetermined percentage of best histogram values. In block 910, the Bezier control points of the non-preferred group of are re-represented inserting an additional on-line control point, inserting an additional off-line control point, deleting an existing on-line control point, deleting an existing off-line control point, perturbing an existing

on-line control point, or perturbing an existing off-line control point. The process then continues to block 420.

FIG. 10 is an exemplary block diagram of an electronic device 1000 according to the present invention. The electronic device may be, but is not limited to, a radio telephone such as a cellular telephone, a personal digital assistant ("PDA"), a hand-held computer, or any computing and/or communicating device. The electronic device 1000 comprises a display 1002, a processor 1004, a memory 1006, and a power supply 1008. The processor 1004 has internal modules including a digital ink stroke generator 1010, a Bezier curve generator 1012, an element separator 1014, a line converter 1016, a modifier 1018, a first order difference calculator 1020, and a data compressor 1022. The power supply 1008 is controlled by the processor 1004 to provide power to the internal components so that they may function properly.

The display 1002 displays a drawn object. Data representing the drawn object may be imported to the electronic device 1000 from another device, or if the display 1002 is a touch pad, a user may draw on the display 1002 to provide the drawn object. The display may also display a resulting object approximating the drawn object based upon the compressed data. The processor 1004 then captures the drawn object on the display 1002 as digital ink. Once the drawn object is captured as digital ink, the digital ink stroke generator 1010 divides the captured digital ink into a series of strokes. The Bezier curve generator 1012 then converts each stroke into a corresponding Bezier curve characterized by Bezier control points. The element separator 1014 evaluates each Bezier curve and separates the Bezier curves into first and second groups based upon a predetermined condition. The first group of Bezier curves, satisfying the first predetermined condition, is deemed to be adequately representable by straight lines, and the line converter 1016 converts the first group of Bezier curves into corresponding straight lines. The second group of Bezier curves is evaluated for further compression. Each Bezier control point has a corresponding element identification, an X-axis coordinate, a Y-axis coordinate, and a curve status, where the X-axis and Y-axis coordinates represent coordinates of the Bezier control point on the display and the curve status indicates whether the Bezier control point is on-line control point or off-line control point. The first order difference calculator 1020 separates the coordinate information into an X-coordinate array and a Y-

coordinate array and stores them into the memory 1006. The first order difference calculator 1020 calculates first order differences between consecutive array elements of each coordinate array to determine whether to perform further data compression. Based upon the calculated results of the first order difference calculator 1020, the modifier 1018 adjusts the existing Bezier control points in several ways to reduce the overall data size. The modifier 1018 may adjust the existing Bezier control points by inserting additional on-line control points, by inserting additional off-line control points, by deleting some of existing on-line control points, by deleting some of existing off-line control points, by perturbing some of existing on-line control points, by perturbing some of existing off-line control points, or by any combination of the above. The data compressor 1022 then losslessly compresses the combined data of the converted first group, which are approximations by straight lines, and the modified second group, which are approximations by modified Bezier curves.

While the preferred embodiments of the invention have been illustrated and described, it is to be understood that the invention is not so limited. Numerous modifications, changes, variations, substitutions and equivalents will occur to those skilled in the art without departing from the spirit and scope of the present invention as defined by the appended claims.